Human Factors in Design, Engineering, and Computing, Vol. 238, 2026, 852–866
https://doi.org/10.54941/ahfe1008317

Developing an Approach to Measure Trust in Flight Testing of Autonomous Aircraft Systems: An LLM-Assisted Robotic Journaling Proof of Concept

Nathalie Ochoa[1], Martin Ha[1], Justin Stofik[2], Rachel Jager[2], Brenda Haslup[2], Boyoung Kim[3], and Nhut Ho[1]

[1]California State University of Northridge, Northridge, CA 91324, USA
[2]Edwards Human Systems Integration Technical Test Force, Edwards, CA 93523, USA
[3]George Mason University Korea, Republic of Korea

ABSTRACT

Integrating autonomous systems into high-stakes military aviation requires robust, repeatable methods for evaluating operator trust. While developmental flight testing generates exhaustive technical log data during envelope expansion, these quantitative records capture system states but fail to document the nuanced cognitive evolution of operator trust. This paper presents a case study for applying the Robotic Journaling methodology originally developed and validated for heterogeneous human-machine teams in R3 (Real users, Real systems, Real consequences) environments to autonomous flight tests. As a precursor to Phase 2 flight testing, this Phase 1 study utilizes simulated data from a software control environment to establish and validate a Trust Calibration Indicator Codebook for flight test maneuvers. The approach uses a Large Language Model (LLM) to aggregate multi-modal data on system behavior, alerts, operator interventions, and communications into a coherent narrative for qualitative analysis. Flight logs contain extensive data about what events took place during a test point; LLMs aggregate that information into a narrative that non-engineers can understand, which can then be correlated to an operator's retrospective account of their experience on that flight. These narratives are then used as the basis for a cross-referencing analysis, a key tool for scoping the boundaries of trust calibration. This study establishes a replicable framework for analyzing how an operator's trust evolves in response to an autonomous system's performance during flight test maneuvers. The central contribution is the Trust Calibration Indicator Codebook: a structured set of observable codes derived from over-trust and under-trust theory that operationalizes trust calibration for flight tests. Phase 2 is positioned to empirically calibrate and validate these indicator codes against actual flight test data.

Keywords: Trust calibration, Robotic journaling, LLM-assisted analysis, Autonomous flight test, Cross-referencing, Over-trust, Under-trust, Human-autonomy teaming, Developmental test & evaluation

INTRODUCTION

The integration of autonomous systems into military aviation is changing how flight test campaigns are planned and executed. As autonomous aircraft are evaluated in increasingly complex developmental and operationally relevant

Received September 26, 2026; Revised November 2, 2026; Accepted November 12, 2026; Available online December 1, 2026

© 2026 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

scenarios, effective human-autonomy coordination becomes critical to safety, test execution, and mission effectiveness. Central to this relationship is trust calibration: the degree to which an operator's confidence in the autonomous system aligns with the system's actual capabilities and limitations.

Developmental flight tests produce technical evidence about autonomous-system performance. Aircraft telemetry, system event logs, alerts, operator inputs, and control room communications can document what occurred during a test point. However, this data fails to capture the nuanced evolution of an operator's trust. Flight test engineers and stakeholders therefore face a significant challenge: they may have extensive evidence of what the system did and how the operator responded, but limited insight into whether the operator's trust was appropriately calibrated. This challenge is compounded by a deeper issue: what an operator recalls feeling vulnerable about during a maneuver often differs from their logged reaction.

This disconnect presents an opportunity to better understand how operator trust develops during flight tests. By cross-referencing recalled vulnerabilities with logged reactions, flight test engineers can identify how an operator's trust evolves. This paper demonstrates a systematic approach to measuring trust during flight testing, using cross-referencing to scope trust boundaries, with a Large Language Model (LLM)-assisted proof of concept using simulated software-based control-environment data and Robotic Journaling as the case-study methodology.

The underlying premise of this approach is that LLMs can ingest and synthesize heterogeneous, multi-modal data streams to facilitate rigorous qualitative analysis. While flight logs provide exhaustive quantitative records of system states, LLMs reconstruct these events into a chronologically indexed, human-readable narrative. This narrative can then be systematically correlated with the operator's retrospective account, forming the operational foundation of the Robotic Journaling methodology as applied to flight tests.

THE TRUST CALIBRATION PROBLEM

Trust calibration in human-autonomy teaming is not a binary state but a dynamic, continuous spectrum. At one end lies over-trust: the operator places excessive confidence in the autonomous system, leading to insufficient monitoring and delayed intervention when the system deviates from expected behavior. At the other end lies under-trust; the operator lacks confidence in the system, resulting in unnecessary interventions, reduced mission effectiveness, and failure to leverage autonomous capabilities.

Both extremes pose significant risks in R3 environments (Real users, Real systems, Real consequences). Over-trust can lead to catastrophic outcomes when an autonomous system fails, and the operator is not prepared to intervene. Under-trust can degrade mission performance and prevent the realization of autonomous system benefits. The goal, therefore, is calibrated trust: an operator's confidence that matches the system's actual reliability and performance characteristics.

Prior research in high-stakes heterogeneous human-machine teams has demonstrated that performance trust, which can be referred to as confidence in a teammate's capability to execute tasks, dominates in operational

contexts, while moral trust, confidence in a teammate's intentions and ethical decision-making, is nearly absent for robotic teammates (Kim et al., 2025). This finding suggests that robots are conceptualized as tools rather than teammates, even when tasked with teammate-like responsibilities. This tool-teammate mismatch has direct implications for autonomous aircraft, where operators must trust the system to execute complex maneuvers while maintaining vigilance for failures.

Measuring trust calibration in flight test environments presents unique challenges. Traditional methods such as surveys, interviews, and subjective ratings are often intrusive, retrospective, and disconnected from the actual system behavior. They capture what operators say about their trust, not necessarily what their behavior reveals about their trust. In high-stakes flight test environments, direct observation may be limited, intrusive, or unavailable altogether.

The Robotic Journaling Methodology

This paper applies Robotic Journaling, an existing methodology originally developed for heterogeneous human-machine teams in R3 environments (Ho et al., 2025), to autonomous flight testing. The methodology uses LLMs to transform complex, raw flight-test log data into structured, human-readable narratives. These narratives serve as the basis for cross-referencing analysis, enabling flight test engineers to scope the boundaries of trust calibration.

The methodology addresses a fundamental gap: flight test logs capture what occurred but fail to capture the cognitive significance of those events. Raw logs contain timestamps, parameter values, and event codes, but lack the semantic context required to explain human behavior. They do not explain why an operator hesitated, why a maneuver was terminated, or why trust was or was not calibrated. By aggregating multi-modal data into coherent narratives, this methodology bridges this gap, contextualizing objective flight data with the operator's subjective experience.

The Robotic Journaling methodology was originally developed and validated in the context of the DARPA Subterranean Challenge (Team CoSTAR) and the Mars 2020 mission (Ho et al., 2025). In those contexts, the method transformed 536 pages of fragmented robot logs from 151 days of field testing into 228 pages of coherent narratives, making field-generated data accessible when direct observation was impossible or insufficient. This paper extends the methodology to autonomous flight testing, demonstrating its applicability to a new R3 domain.

The generated narratives can then be correlated with an operator's retrospective account. This cross-referencing reveals discrepancies between recalled and logged trust indicators, discrepancies that are themselves valuable data for understanding trust calibration. For example, an operator who recalls feeling trust, but whose logged behavior shows repeated unnecessary manual interventions may be exhibiting unconscious under-trust. Conversely, an operator who recalls feeling vulnerable, but whose logged behavior shows appropriate monitoring without unnecessary interventions, may be exhibiting well-calibrated trust in action.

Trust Calibration Indicator Codes

The central contribution of this paper is the development of a Trust Calibration Indicator Codebook, a structured set of observable codes that operationalize over-trust and under-trust theory for flight test. These codes are derived from established trust calibration frameworks (Lee and See, 2004) and empirical findings from high-stakes HMT research (Kim et al., 2025).

Table 1: The codebook organizes trust calibration indicators into three primary categories: over-trust, under-trust, & calibrate.

CALIBRATED TRUST (CT)	OVER-TRUST (OT)	UNDER-TRUST (UT)
Operational Definition: Optimal calibration between an operator's trust and the actual capabilities and limitations of the automated system, leading to effective human-automation collaboration.	Operational Definition: An operator's excessive trust in the automation relative to its actual reliability and performance, resulting in complacency and potential for automation-related errors.	Operational Definition: An operator's lack of trust in the automated system despite its capability to perform tasks effectively, leading to reduced usage and loss of potential benefits.
INDICATOR CODES:	**INDICATOR CODES:**	**INDICATOR CODES:**
• CT-1 Appropriate Watch	• OT-1 Missed Watch	• UT-1 Premature Terminate
• CT-2 Timely Intervention	• OT-2 Delayed Intervention	• UT-2 Excessive Monitoring
• CT-3 Appropriate Monitoring	• OT-3 Insufficient Monitoring	• UT-3 Unnecessary Intervention
• CT-4 Confidence in Recovery	• OT-4 Complacency	• UT-4 Failure to Leverage

Over-trust (OT): Operator places excessive confidence in the system, leading to insufficient monitoring. **Under-trust (UT):** Operator lacks confidence in the system, leading to unnecessary interventions. **Calibrated Trust (CT):** Operator monitoring and intervention match system performance and task demands.

GROUND RULES AND ASSUMPTIONS

The following ground rules and assumptions establish the operational context for this study. These constraints matter because trust calibration is context-dependent: an operator's confidence in an autonomous system cannot be evaluated in the abstract, but only relative to the specific conditions, authority structures, and risk tolerances of the test environment. In developmental flight testing, operators are not typical end-users but trained evaluators whose interventions (or lack thereof) carry safety-of-flight implications. Similarly, the control room's role in authorizing,

monitoring, and terminating maneuvers creates a distributed trust network that differs from single-operator laboratory paradigms. By specifying these ground rules upfront, we ensure that the Trust Calibration Indicator Codebook reflects the actual decision ecology in which autonomous aircraft are tested, rather than idealized assumptions about how operators interact with automation.

Table 2: Ground rules and assumptions.

GROUND RULE / ASSUMPTIONS, OPERATIONAL CONTEXT, & IMPACT ON STUDY & TRUST CALIBRATION

GROUND RULE/ASSUMPTION Category	OPERATIONAL CONTEXT How the System Functions	IMPACT ON STUDY & TRUST CALIBRATION Significance for Evaluation
Autonomous Aircraft w/ Person on the Loop	The operator monitors autonomous actions, receives advance notifications of upcoming maneuvers, and can intervene or take manual control as needed	Requires constant monitoring and situational awareness. Trust calibration is critical to ensure operators neither over-rely on the system nor intervene unnecessarily
Trust Measurement Objective	Measures human trust continuously throughout the evaluation to extract actionable insights	Generates empirical data to inform the flight test community, acquisition programs, and end users regarding operator confidence
Envelope Expansion Mission	Flight operations focus on gathering aerodynamic data to validate predictive aero models and expand the platform's safe operating limits.	Flight logs provide extensive technical performance metrics, but trust metrics must be gathered concurrentty to capture the operator's psychological response.
Command Control Room	A control room team actively monitors vehicle safety of flight (SoF) and safety of test (SoT) parameters.	Control room communication loops and intervention triggers offer rich verbal and behavioral data for calibration analysis.
Operator Expertise	Evaluators are domain experts who are also extensively trained to evaluate the platform from the perspective of typical field users.	Provides highly nuanced trust calibration data that bridges elite flight-test evaluation with practical operational utility
Engineering Command	Autonomous maneuvers are pre-planned and executed vie standardized engineering command structures	Guarantees consistent and reproducible maneuver execution, establishing a baseline to laclate operator trust as a variable

ANALYZING HUMAN-MACHINE TRUST IN AUTONOMOUS SYSTEMS

SCENARIO BACKGROUND

The scenario involves an autonomous aircraft with a person-in-the-loop operating in a developmental flight test environment. The operator can take control of the aircraft and is notified of autonomous actions prior to execution. The envelope expansion mission aims to validate aero models and expand the safe flight envelope. Operators are experts trained to evaluate the system for the typical user in the field. Using an engineering command or mission planning, the autonomous aircraft executes pre-planned maneuvers. The control room monitors the air vehicle for safety of flight and safety of test parameters. The DE, referring to the evaluating authority, is responsible for judging whether a maneuver was performed correctly.

The notional flow for a typical test point proceeds as follows:

CTRL: briefs conditions and limits;	**TEST**: confirms on conditions and ready to maneuver;	**DE**: calls good maneuver or requests repeat;
CTRL and TEST: confirm configuration;	**CTRL** clears TEST to maneuver;	**CTRL:** relays point complete, or requests repeat;
CTRL: confirms room ready.	**TEST**: calls completed maneuver.	**CTRL**: briefs next maneuver.
CTRL: briefs maneuver deck.		

Standard communications like "CTRL briefs conditions and limits," "CTRL clears TEST to maneuver," and "TEST calls completed maneuver" provide clear indicators of coordination and trust calibration between control and test teams. The sequence of brief, confirm, clear, execute, report, and evaluate ensures every maneuver is authorized, observed, and validated before proceeding. Calls such as "DE calls good maneuver or requests repeat" and "CTRL relays point complete" confirm that results are assessed and either accepted or redone.

Here's a short explanation in that same style:

The following standard communications provide valuable indicators of trust calibration: "Point complete" (execution of the stated maneuver is complete), "Watch (parameter)" (test conditions may be in danger of being exceeded), "Terminate" (interrupt and stop execution), and "Knock it off, knock it off, knock it off" (safety of flight issue, cease maneuvering).

In short: These calls form a disciplined loop of brief → confirm → clear → execute → report → evaluate → repeat or advance; ensuring every maneuver is authorized, observed, and validated before moving on.

MANEUVER DESCRIPTIONS

Trim Shot



- **Aircraft stabilizes in all axes at the specified conditions with no lateral load factor using pitch and roll trim only.**
- **The operator calls "Ready, Read, Now" when the engineering function is activated.**
- **Fly in this trim condition for 30 seconds.**

^Altitude deviations are allowed ± 50 ft. from Operator call of "Ready."

^Airspeed deviations are allowed ± 5 KIAS from Operator call of "Ready."
~ Operator and Control Room monitor duration of trim shot.

~ The trim shot provides an opportunity to observe trust calibration in a relatively low workload maneuver.

Operators who trust the autonomous system appropriately will allow it to maintain trim without unnecessary intervention.

Operators who exhibit under trust may intervene unnecessarily, while those who exhibit over trust may fail to monitor for deviation

Pitch, Roll, and Yaw Doublets

Aircraft trim at the specified conditions/configuration

Pitch Doublet: 25–30% square stick inputs that last 0.5 seconds for each side of the doublet (aft stick first). Allow the aircraft to respond with no inputs for 5 seconds or until stable. Recover to wings level and restabilize for next doublet or maneuver.	**Roll Doublet:** 25–30% square stick inputs that last 1.5 seconds for each side of the doublet (left stick first). Allow the aircraft to respond with no inputs for 5 seconds or until stable. Recover to wings level and restabilize for the next doublet or maneuver.	**Yaw Doublet:** 25–30% square pedal inputs that last 2 seconds for each side of the doublet (left pedal first). Allow the aircraft to respond with no inputs for 5 seconds or until stable. Recover to wings level and restabilize for the next doublet or maneuver

Doublets are dynamic maneuvers that require the operator to monitor the autonomous system's response and intervene if necessary. These maneuvers are particularly revealing for trust calibration because they involve rapid changes in aircraft state and require the operator to decide whether the autonomous response is appropriate.

METHODOLOGY: ROBOTIC JOURNALING APPLIED TO FLIGHT TEST

Robotic Journaling is an existing methodology (Ho et al., 2025) that uses Large Language Models to transform complex, raw flight-test log data into structured, human-readable narratives. The methodology consists of four primary stages: **(1) Data Aggregation, (2) Narrative Generation, (3) Cross-Referencing Analysis, and (4) Trust Calibration Assessment.**

The proposed system uses a locally hosted LLM to transform timestamped, multi-modal flight-test data into structured narratives for trust-calibration analysis. For this specific case study, we prefer a local model because flight-test data may contain sensitive technical information, operational procedures, and operator communications that should not be transmitted to external cloud services.

The system is intended to operate as an offline or closed-network decision-support tool. It does not control the aircraft, issue flight commands, or replace the operating control room personnel. Its purpose is to organize and summarize flight-test data for later analysis.

Data Aggregation and Timestamp Normalization

Flight test logs contain extensive data about events that occurred during test point. This data includes system behavior (autonomous system actions, maneuver execution, parameter values), alerts (system-generated notifications, warnings, and cautions), operator interventions (control inputs, terminations, and other operator actions), and communication patterns (verbal communications between operator, control room, and other stakeholders). In Phase 1, this data is simulated from a software control environment. In Phase 2, the aim is to use actual flight test data.

Table 2: The system collects data from multiple sources, including aircraft telemetry, autonomoussystem events, alerts, operator inputs, controlroom communications, and test-point metadata.	**Table 2: Each record is associated with a timestamp. These timestamps are converted to a common time reference so that events from different sources can be accurately compared.**

For example, the system may align the following events:

Table 2: Data aggregation and timestamp normalization.

[10:15:23.000]	TEST:	Ready, Read, Now
[10:15:23.120]	SYS:	Trim function activated
[10:15:30.000]	SYS:	Altitude = 10,025 ft
[10:15:40.000]	TEST:	Watch Altitude
[10:15:45.000]	SYS:	Altitude = 10,052 ft
[10:15:53.000]	TEST:	Point Complete

Table 2: The system preserves the original data while creating a separate analysis copy. This allows every statement in the generated narrative to be traced back to its source. In Phase 1, this data is simulated from a software control environment. In Phase 2, the aim is to use actual flight test data.

Multi-Modal Aggregation and Structured Event Representation

After normalizing timestamps, the system combines the data into a chronological event timeline. Telemetry may be recorded at a high frequency, while alerts, operator actions, and communications occur only at specific moments. The aggregation process therefore groups relevant data within defined time windows surrounding important events.

For example, a warning issued at 10:15:40 may be associated with telemetry recorded several seconds before and after the warning. This allows analysts to determine what the aircraft was doing before the event, whether the system was within its expected limits, whether an alert was generated, and how quickly the operator responded.

The aggregated information is converted into a structured representation before it is provided to the LLM. Each event may include its timestamp, source, event type, parameter, value, and test-point identifier.

Local LLM Narrative Generation

The structured event timeline is provided to a locally hosted LLM. The LLM generates a human-readable narrative describing system behavior, alerts, operator monitoring, interventions, and test-point completion. These narratives describe maneuver execution, autonomous system performance, operator monitoring, alerts, tolerance exceedances, and communication patterns. The narrative format makes the data accessible to non-engineers, enabling cross-functional analysis and shared understanding.

The LLM is instructed to use only the supplied flight-test data, preserve timestamps and numerical values, distinguish confirmed events from missing information, and avoid inventing explanations or operator motivations. When possible, each narrative statement references the event records supporting it.

For example, the generated narrative may state:

"The operator initiated the trim shot at 10:15:23. The autonomous system activated the trim function and maintained altitude and airspeed near the briefed values. At 10:15:40, the operator called 'Watch altitude' after a small altitude deviation. The system subsequently remained within the approved tolerance, and the operator called 'Point complete' at 10:15:53. No manual intervention was recorded."

This narrative makes technical data easier to understand for flight-test engineers, program personnel, and other stakeholders who may not work directly with raw telemetry.

Cross-Referencing Analysis

Cross-referencing analysis correlates generated narratives with operator-reported experiences. This correlation reveals discrepancies between what operators recall feeling vulnerable about and what their logged reactions show. For example, an operator might recall feeling confident during a maneuver, but the logged data might show multiple unnecessary interventions indicating under-trust. Conversely, an operator might recall feeling concerned about a maneuver, but the logged data might show no interventions despite system deviations, indicating over-trust.

Trust Calibration Assessment

The final stage involves applying the Trust Calibration Indicator Codebook to cross-reference results. The codebook is applied to cross-referenced narratives. For each maneuver or test point, the analyst reviews the narrative and operator interview data, then assigns codes based on observable indicators. Multiple codes may apply to a single maneuver.

For Example:

A trim shot where the operator calls "Ready, Read, Now," the system maintains trim within tolerances for 30 seconds, and the operator calls "Point complete" would receive codes CT-1 (Appropriate Watch), CT-3 (Appropriate Monitoring), and CT-4 (Confidence in Recovery). A pitch doublet where the system exceeds the pitch rate target by 15% and the operator calls "Terminate" at 2 seconds would receive codes UT-1 (Premature Terminate) and UT-3 (Unnecessary Intervention).

By applying the codebook across multiple test points, patterns of trust calibration emerge. Trust Growth is indicated when CT codes increase over time, and UT codes decrease. Over-trust risk is indicated when OT codes increase over time, and CT codes decrease. Under-trust persistence is indicated when UT codes remain high, and CT codes remain low. Boundary Sensitivity is indicated when codes shift at performance limits.

System Architecture and Safety Controls

Because flight-test information may be sensitive, the proposed system operates on a local computer, secure test-network server, or other approved closed environment. Raw logs, communications, and generated narratives are not transmitted to external APIs unless explicitly authorized.

The system includes local model execution, access controls, encryption for stored data, audit trails, and clear separation between the analysis system and aircraft-control systems. The LLM does not issue flight commands or automatically alter aircraft behavior. The LLM is treated as an analysis assistant rather than an authority. Its output may contain errors and must be reviewed against the original flight-test logs before trust codes are finalized.

CASE STUDY: SIMULATED FLIGHT TEST DATA

In Phase 1, simulated data from a software control environment is used to develop and validate the Robotic Journaling methodology and Trust Calibration Indicator Codebook. The simulated environment replicates the types of data generated during actual flight test campaigns, including system behavior logs, alert and warning logs, operator intervention logs, and communication transcripts. **The following example illustrates how Robotic Journaling transforms raw log data into a human-readable narrative.**

Raw Log Data Shows:

[10:15:40] TEST: Watch altitude

[10:15:23] TEST: Ready, Read, Now	[10:15:23] SYS: Trim function activated	[10:15:25] SYS: Pitch trim: 2.3 deg, Roll trim: -1.1 deg	[10:15:30] SYS: Altitude: 10,025 ft, Airspeed: 202 KIAS	[10:15:35] SYS: Altitude: 10,048 ft, Airspeed: 203 KIAS	[10:15:45] SYS: Altitude: 10,052 ft, Airspeed: 204 KIAS	[10:15:50] SYS: Altitude: 10,038 ft, Airspeed: 202 KIAS	[10:15:53] TEST: Point complete

The generated narrative reads:

"The operator initiated the trim shot at 10:15:23 with the call 'Ready, Read, now.' The autonomous system activated the trim function and established pitch trim at 2.3 degrees and roll trim at –1.1 degrees. Over the next 30 seconds, the aircraft-maintained altitude within ±52 ft of the briefed 10,000 ft and airspeed within ±4 KIAS of the briefed 900 KIAS. At 10:15:40, the operator called 'Watch altitude' when the aircraft reached 10,048 ft. The autonomous system corrected the deviation, and the operator called 'Point complete' at 10:15:53."

The generated narrative is then cross-referenced with operator-reported experiences. The operator interview states: "I felt confident during the trim shot. The system seemed to be holding the trim well, and I didn't feel the need to intervene." The cross-reference analysis shows that the operator's reported confidence aligns with the logged data. The operator monitored the system appropriately, called "Watch altitude" when the aircraft deviated, and allowed the autonomous system to correct the deviation without intervention. This pattern suggests calibrated trust, with codes CT-1, CT-3, and CT-4 applied.

Across multiple test points and maneuvers, patterns of trust calibration emerge: Calibrated trust: Operator monitors appropriately, intervenes when necessary, and allows the system to perform within tolerances. Over trust: Operator fails to call "Watch" when parameters deviate, or delays intervention when system performance degrades. Under trust: Operator intervenes unnecessarily, terminates maneuvers prematurely, or fails to leverage autonomous capabilities. This pattern suggests that trust calibration varies by maneuver type and evolves. The codebook enables systematic tracking of these changes.

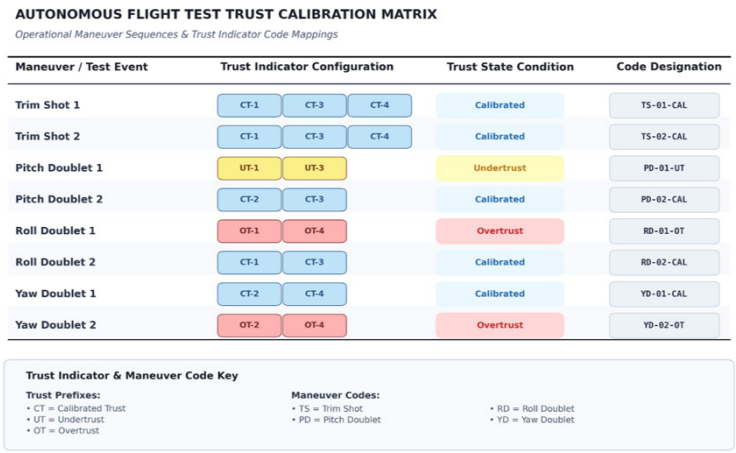


Figure 1: Example code applications across a simulated campaign.

Figure 1: Distribution of operator trust calibration states and corresponding indicator code configurations across developmental flight test maneuvers.

PHASE 2: THE AIM TOWARDS CALIBRATING THE INDICATOR CODES

Phase 2 aims to produce a validated, calibrated Trust Calibration Indicator Codebook ready for operational use in flight test campaigns. Specific objectives include refining code definitions based on actual flight test data, splitting, merging, or eliminating codes based on their utility in practice and establishing inter-rater reliability by having multiple analysts apply the codebook to the same data and calculating agreement statistics such as Cohen's kappa. Utility in practice is judged empirically; a code earns its place only if analysts can apply it reliably, it occurs often enough to matter, it captures something distinct from other codes rather than overlapping with them, and it adds explanatory value by distinguishing well calibrated from poorly calibrated cases-codes that fail these tests are split, merged, or eliminated. Codes will also be validated against external criteria including operator self-reports of trust, performance outcomes such as maneuver success, and control room assessments of operator trust.

Sensitivity analysis will determine whether codes detect subtle changes in trust calibration, with code refinement as needed to capture nuanced

behaviors. Longitudinal validation will track code patterns across an entire test campaign to confirm the predicted patterns of Trust Growth, over trust Risk, under trust Persistence, and Boundary Sensitivity. Thresholds will be established for what constitutes high versus low code frequencies; for example, how many UT-1 codes per test point indicate problematic under trust. Ultimately, Phase 2 will explore whether LLMs can automatically apply codes to narratives, aiding analysts and increasing scalability.

System Evaluation

The system is evaluated using both technical and human-centered measures. Technical evaluation examines whether the LLM preserves timestamps, numerical values, event order, and operator actions. Human evaluation determines whether analysts can use the generated narratives to identify trust indicators consistently.

Evaluation measures include narrative accuracy, timestamp accuracy, event-order accuracy, analyst agreement, reduction in review time, and false-positive or false-negative trust indicators. Inter-rater reliability measures, such as Cohen's kappa, are used to determine whether different analysts apply the codebook consistently.

Implications for Flight Test

The Robotic Journaling methodology and Trust Calibration Indicator Codebook have significant implications for flight tests. By making flight test logs interpretable and analyzable, the method supports evidence-based assessment of trust calibration. This enables flight test engineers to identify trust-related patterns that may not be apparent from raw logs alone, correlate technical system behavior with operator-facing indicators, assess trust evolution over time across multiple test points, and inform test processes, system design, and operator training. Rather than relying solely on subjective assessments or retrospective interviews, flight test engineers can now use the narratives generated through Robotic Journaling to ground trust discussions in observable evidence. This represents a shift from asking operators how they felt about a maneuver to examining what their behavior revealed about their trust. The codebook provides a shared vocabulary for these discussions, enabling engineers, operators, and program stakeholders to communicate about trust using consistent, observable indicators. Over the course of an envelope expansion campaign, this approach allows for longitudinal tracking of trust calibration, revealing whether trust grows appropriately with experience or whether problematic patterns of over trust or under trust emerge and persist.

Implications for Human-Autonomy Teaming

The methodology also has implications for human-autonomy teaming more broadly. By providing a structured approach to measuring trust calibration, the method supports the design of autonomous systems that foster calibrated trust, the development of training programs that address trust calibration,

and the assessment of trust in operational environments beyond flight test. Prior research in high-stakes heterogeneous human-machine teams has shown that performance trust dominates while moral trust is limited for robots (Kim et al., 2025). This finding suggests that operators may conceptualize autonomous aircraft as tools rather than teammates even when those aircraft are tasked with teammate-like responsibilities. The Trust Calibration Indicator Codebook can help identify when this tool-teammate mismatch leads to trust calibration failures. For example, if an operator consistently fails to monitor an autonomous system because they perceive it as a deterministic tool, the codebook will capture this as a pattern of over trust codes (OT-1, OT-3, OT-4). Conversely, if an operator repeatedly intervenes unnecessarily because they do not trust the system's judgment, the codebook will capture this as a pattern of under trust codes (UT-1, UT-3). By making these patterns visible, the methodology enables designers and trainers to address the underlying mismatch between how autonomous systems are perceived and how they are used.

LIMITATIONS

A primary limitation of this study is the use of simulated rather than actual flight test data. While simulated data captures the structure and types of events found in real logs, it may not fully replicate the complexity, noise, or edge cases present in operational data. Real flight test logs contain nuances unexpected sensor behavior, ambiguous communication, overlapping events that simulated data cannot fully reproduce. This limitation means that the codebook's applicability to real-world data remains untested. Some codes may prove difficult to apply when logs are incomplete or when operator communications are sparse. Additionally, the Trust Calibration Indicator Codebook is proposed as a proof of concept. The codes have not yet been validated against real flight test data, and their relationship to actual trust outcomes such as mission success, safety incidents, or operator self-reports remains unestablished. Without this validation, the codebook should be viewed as a hypothesis about how trust manifests in flight tests rather than a confirmed measurement tool. Phase 2 will address these limitations by applying the methodology to real flight test logs, refining code definitions based on operational experience, and validating codes against external criteria.

FUTURE WORK

Future work (Phase 2) will involve applying the Robotic Journaling methodology and Trust Calibration Indicator Codebook to actual flight test data from autonomous aircraft systems. This will allow for validation of the approach in an operational context and further refinement of the trust calibration indicators identified in this proof of concept. Specific Phase 2 objectives include refining code definitions based on real data, establishing inter-rater reliability among multiple analysts, validating codes against external criteria such as operator self-reports and performance outcomes, conducting sensitivity analysis to determine whether codes detect subtle

changes in trust, tracking code patterns over an entire test campaign to validate predicted patterns (Trust Growth, Over-trust Risk, Under-trust Persistence, Boundary Sensitivity), establishing thresholds for what constitutes high versus low code frequencies, and exploring whether LLMs can automatically apply codes to narratives to reduce analyst burden. Additional future work includes extending the methodology to multi-aircraft and multi-operator scenarios, integrating real-time trust calibration monitoring into flight test operations, developing domain-specific LLM prompts for different types of maneuvers and systems, and validating the methodology across different aircraft platforms and mission types. As autonomous systems become increasingly prevalent in military aviation, the ability to measure and calibrate trust in operational contexts will become not just an academic question but an operational necessity.

CONCLUSION

This paper presents a proof-of-concept application of Robotic Journaling for measuring trust calibration in flight testing of autonomous aircraft systems. By leveraging Large Language Models to aggregate multi-modal data and transform complex, raw flight test log data into structured, human-readable narratives, the method bridges the gap between technical system performance and operator trust.

The central contribution is the Trust Calibration Indicator Codebook; a structured set of observable codes that operationalize calibrated trust, over-trust, and under-trust theory for flight test. The codebook aims to enable systematic assessment of trust across test points and maneuvers, providing a shared vocabulary for engineers, operators, and program stakeholders.

The study establishes a replicable framework for analyzing how an operator's trust evolves in response to an autonomous system's performance during flight test maneuvers. By using logged system behavior, alerts, and operator interventions as evidence, the method enables evidence-based assessment of trust calibration grounded in observable data rather than subjective recollection alone.

The anticipated outcome is a repeatable, domain-agnostic framework that enhances transparency and shared understanding, ultimately supporting the convergence of developmental and operational test perspectives. Phase 2 is positioned to empirically calibrate and validate the indicator codes against real flight test data. As autonomous systems become increasingly prevalent in military aviation, methods like Robotic Journaling and frameworks like the Trust Calibration Indicator Codebook will be essential for ensuring safe and effective human-autonomy teaming in R3 environments.

ACKNOWLEDGMENT

Acknowledgements go out to every author, their affiliated institutions and stakeholders, along with previous cohorts who have contributed to making this work possible.

REFERENCES

Hamm, J. A., Möllering, G., & Darcy, K. (2024). Integrating focal vulnerability into trust research. *Journal of Trust Research, 14*(2), 237–255.

Hancock, P. A., Billings, D. R., Schaefer, K. E., Chen, J. Y. C., de Visser, E. J., & Parasuraman, R. (2011). A meta-analysis of factors affecting trust in human-robot interaction. *Human Factors, 53*(5), 517–527.

Ho, N. T., Morrell, B., Kaufmann, M., Kim, B., Ochoa, N., Ha, M., Sagaran, E., & Jannone, J. (2025). Robotic journaling: Translating raw robot logs into structured narratives to model trust in heterogeneous human-machine teams. *AHFE International*.

Kim, B., Jannone, J., Rodriguez, Z. L., Huerta, R., Sagaran, E., Ochoa, N., Shubina, D., Morrell, B., Milano, M., Kaufmann, M., & Ho, N. T. (2025). Performance trust in high-stakes heterogeneous human-machine teams: Insights from interviews with Team CoSTAR and Mars 2020. IEEE RO-MAN

Lee, J. D., & See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors, 46*(1), 50–80.

Mayer, R. C., Davis, J. H., & Schoorman, F. D. (1995). An integrative model of organizational trust. *Academy of Management Review, 20*(3), 709–734.